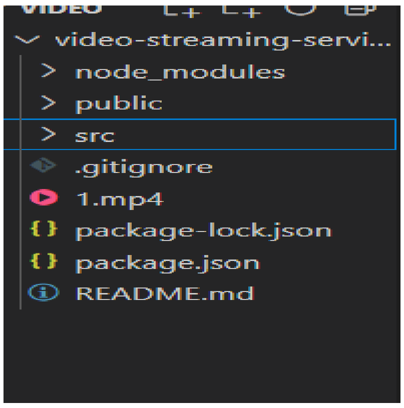
Video Streaming AWS project

Front End:

- **Editor Tool:** VS Code
- **Languages:** React JS, Java script, HTML, CSS

Creating React APP

Step 1: Create One Folder – Give Your Own name
Step 2: Open VS Code – File- Open Folder- Select your Folder.
Step 3: In VS Code – Open terminal:
Type Command:
```
npx create-react-app video-streaming-service
cd video-streaming-service
npm start
```
Step 4: You created folder automatically generated.



Step 5: *In **src and public folder** you can add some **images or logo** for your usage to acceces the image in **App.js file***

Step 6: In *src* folder **delete** Logo , **apptest.js** and other **unwanted** files.
In file **App.js Paste Below Given Code**

```
import React, { useRef, useState } from "react";
import "./App.css";

function App() {
  const videoRef = useRef(null);
  const [isPlaying, setIsPlaying] = useState(false);
  const [currentTime, setCurrentTime] = useState(0);
  const [duration, setDuration] = useState(0);

  const handlePlayPause = () => {
    if (videoRef.current.paused) {
      videoRef.current.play();
      setIsPlaying(true);
    } else {
      videoRef.current.pause();
      setIsPlaying(false);
    }
  };

  const formatTime = (time) => {
    const minutes = Math.floor(time / 60);
    const seconds = Math.floor(time % 60);
    return `${minutes}:${seconds < 10 ? "0" : ""}${seconds}`;
  };

  const handleTimeUpdate = () => {
    setCurrentTime(videoRef.current.currentTime);
    setDuration(videoRef.current.duration);
  };

  return (
    <div className="App">
      <div className="header">
        <img src="a.png" alt="Logo" className="logo" />
        <h2 className="heading">Welcome to My Streaming App</h2>
        <div></div>{" "}
        {/* Placeholder for additional header content like user profile */}
      </div>

      <div className="video-container">
        <video
          controls
          className="video-frame"
          ref={videoRef}
          onClick={handlePlayPause}
          onPlay={() => setIsPlaying(true)}
          onPause={() => setIsPlaying(false)}
          onTimeUpdate={handleTimeUpdate}
        >
          <source src="https://d21rr2y0u23ont.cloudfront.net/1.mp4" type="video/mp4"
/>
          Your browser does not support the video tag.
        </video>
        <div className="video-controls">
          <button onClick={handlePlayPause}>
            {isPlaying ? "Pause" : "Play"}
          </button>
          <div>
            Time: {formatTime(currentTime)} / {formatTime(duration)}
          </div>
          <button>Full Screen</button>
        </div>
      </div>

      <footer className="footer">
        <p>&copy; 2024 Video Streaming App</p>
      </footer>
    </div>
  );
}

export default App;
```

Step 7: In App.css You can create Your Own style or use below code

```
/* App.css */

body {
  margin: 0;
  padding: 0;
  font-family: "Segoe UI", Tahoma, Geneva, Verdana, sans-serif;
  background: linear-gradient(
    to bottom right,
    #1a2a6c,
    #b21f1f,
    #fdbb2d
  ); /* Gradient background */
  background-size: cover;
  background-attachment: fixed;
  min-height: 100vh;
  display: flex;
  justify-content: center;
  align-items: center;
  color: #fff; /* Text color for contrast */
  background-image: url("./6.png");
}

.App {
  width: 100%;
  max-width: 800px;
  background-color: rgba(
    0,
    0,
    0,
    0.8
  ); /* Semi-transparent black background for content */
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.4); /* Shadow for depth */
  padding: 20px;
  border-radius: 8px;
```

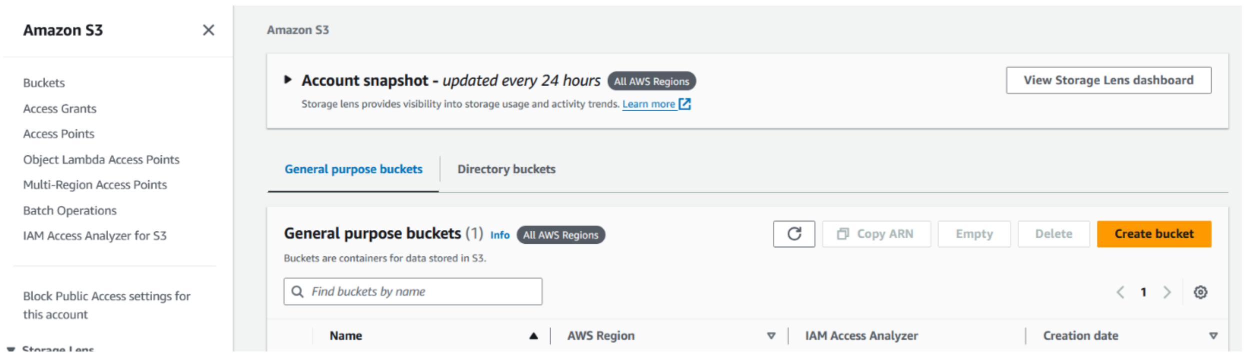
```
}

.header {
  display: flex;
  align-items: center;
  justify-content: space-between;
  margin-bottom: 20px;
}

.logo {
  width: 50px; /* Adjust size as needed */
  height: auto;
}

.heading {
  font-size: 2.5rem;
  font-weight: bold;
  text-align: center;
  margin: 20px 0;
}

.video-container {
  position: relative;
  width: 100%;
  padding-top: 56.25%; /* Aspect ratio for 16:9 video */
  overflow: hidden;
  background-color: #000; /* Black background for video area */
  border-radius: 8px;
  box-shadow: 0 4px 8px rgba(0, 0, 0, 0.4);
}

.video-frame {
  position: absolute;
  width: 100%;
  height: 100%;
  top: 0;
  left: 0;
}

.video-controls {
  position: absolute;
  bottom: 0;
  left: 0;
  width: 100%;
  background-color: rgba(
    0,
    0,
    0,
    0.7
  ); /* Semi-transparent black background for controls */
  display: flex;
  justify-content: space-between;
  align-items: center;
  padding: 8px;
}

.video-controls {
  position: absolute;
  bottom: 0;
  left: 0;
  width: 100%;
  background-color: rgba(
    0,
    0,
    0,
    0.7
  ); /* Semi-transparent black background for controls */
  display: none; /* Initially hide controls */
  justify-content: space-between;
  align-items: center;
  padding: 8px;
}

.video-container:hover .video-controls {
  display: flex; /* Show controls on hover */
}

.footer {
  text-align: center; /* Center align text */
  padding: 5px 0; /* Adjust padding for smaller size */
  color: #f1c0c0; /* Text color */
  background-color: rgba(0, 0, 0, 0.5); /* Semi-transparent black background */
  position: fixed; /* Fixed position at the bottom */
  width: 100%; /* Full width */
  bottom: 0; /* Stick to the bottom */
  left: 0; /* Center alignment */
}
```

Backend AWS Management Console Steps

Step 1: Login to ***AWS Management Console.***

Step 2: Search ***S3*** and Open it.

Step 3: In ***S3***, Click, ***Create Bucket***.
Then in ***General Configuration.***



Amazon S3

Buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3

Block Public Access settings for this account

Amazon S3

Account snapshot - *updated every 24 hours* All AWS Regions
Storage lens provides visibility into storage usage and activity trends. Learn more

View Storage Lens dashboard

General purpose buckets | Directory buckets

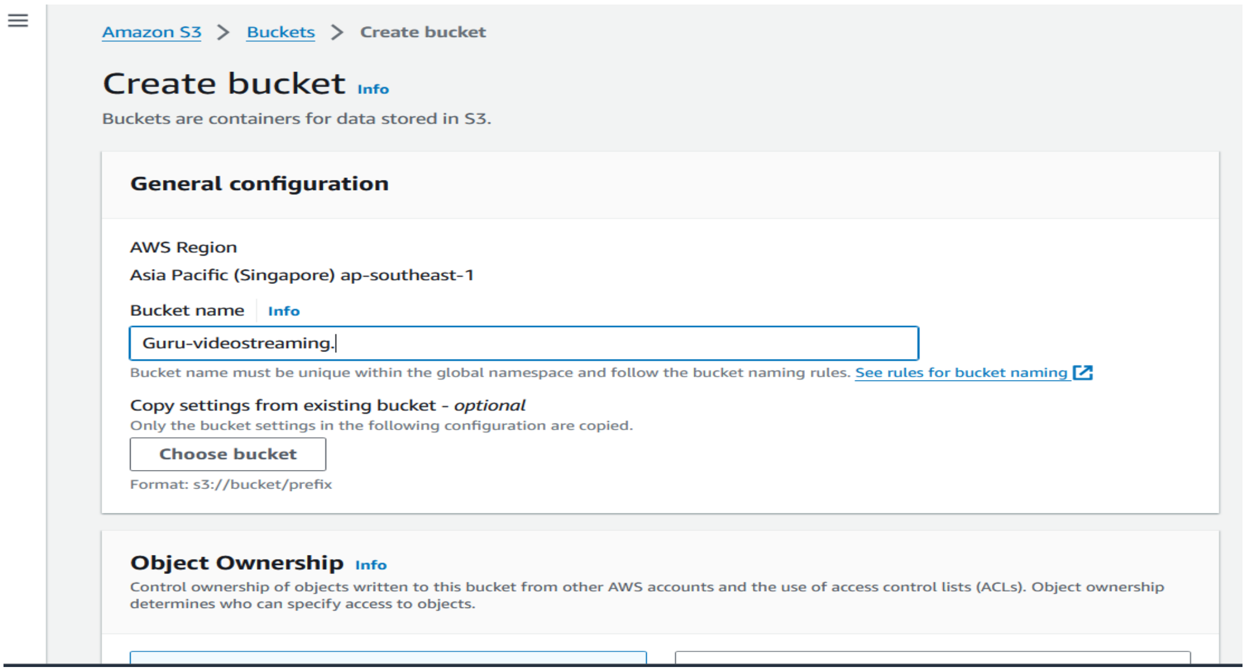
General purpose buckets (1) Info All AWS Regions
Buckets are containers for data stored in S3.

Copy ARN | Empty | Delete | Create bucket

Find buckets by name

Name	AWS Region	IAM Access Analyzer	Creation date

- In the ***Bucket Name*** Field – Give one name
  - example: ***Guru-videostreaming***.



Amazon S3 > Buckets > Create bucket

Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region
Asia Pacific (Singapore) ap-southeast-1

Bucket name Info
Guru-videostreaming.
Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming

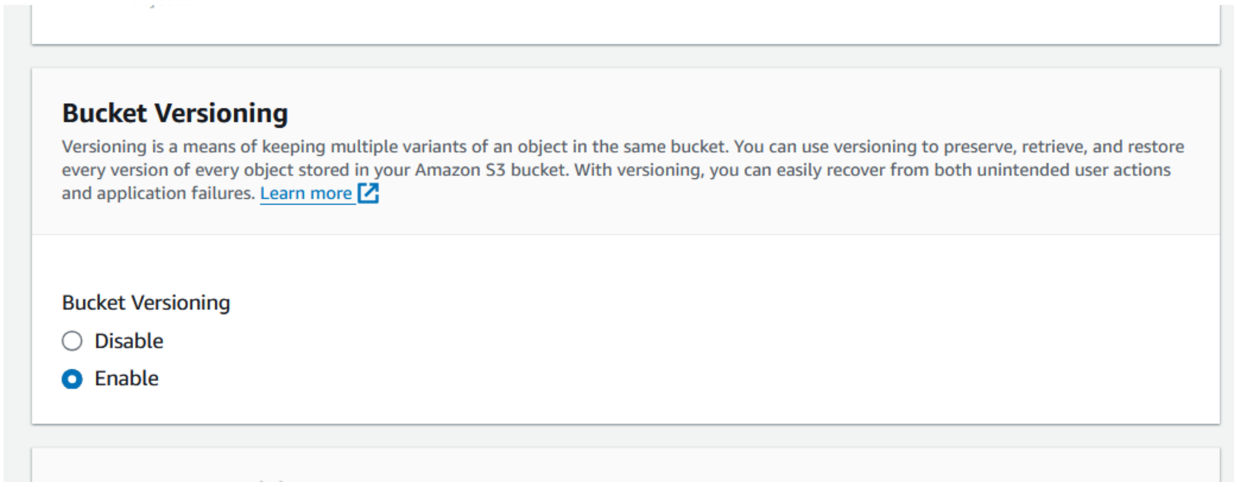
Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.
Choose bucket
Format: s3://bucket/prefix

Object Ownership Info
Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

- In the ***AWS Region*** Field, keep it default,
- In Case your region is not in that select yours.

- Next, keep as it is ***Object Ownership*** and
- ***Block all public access.***

- In the field, ***Bucket Versioning,*** Select ***Enabled.***



Bucket Versioning

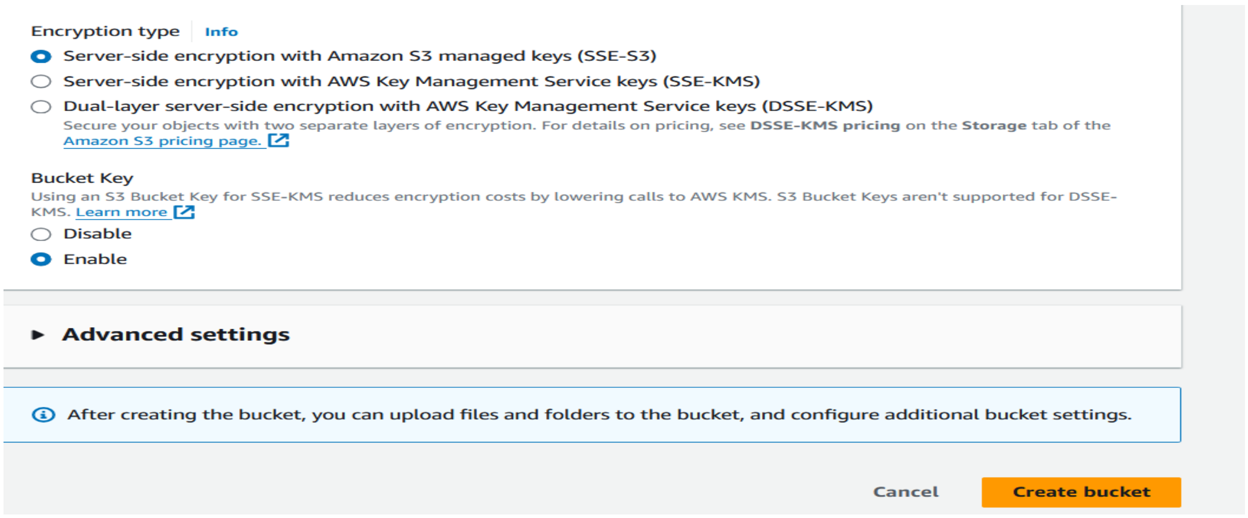
Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures. Learn more

Bucket Versioning

[ ] Disable

[✓ Enable]

- In the ***Default Encryption*** Field, Select ***Enabled***.



Encryption type Info

[✓ Server-side encryption with Amazon S3 managed keys (SSE-S3)]

[ ] Server-side encryption with AWS Key Management Service keys (SSE-KMS)

[ ] Dual-layer server-side encryption with AWS Key Management Service keys (DSSE-KMS)
Secure your objects with two separate layers of encryption. For details on pricing, see **DSSE-KMS pricing** on the **Storage** tab of the Amazon S3 pricing page.

Bucket Key

Using an S3 Bucket Key for SSE-KMS reduces encryption costs by lowering calls to AWS KMS. S3 Bucket Keys aren't supported for DSSE-KMS. Learn more

[ ] Disable

[✓ Enable]

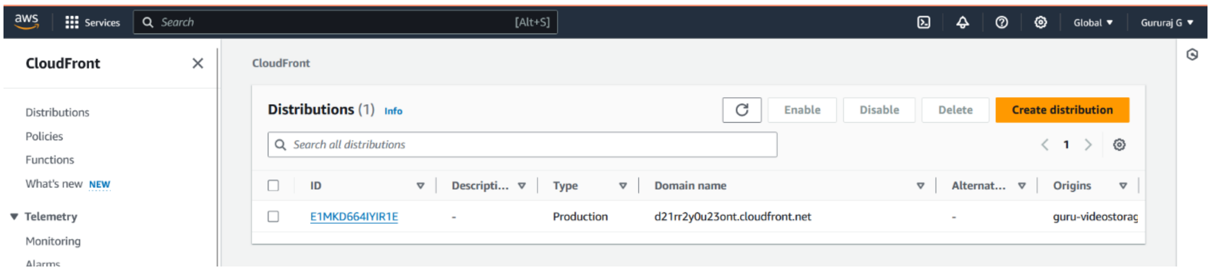
▸ **Advanced settings**

After creating the bucket, you can upload files and folders to the bucket, and configure additional bucket settings.

Step 4: Select, ***Create Bucket.***

Finally, Bucket is Successfully Created.

Step 5: Search ***Cloud Front*** and Open it in ***New Tab***.

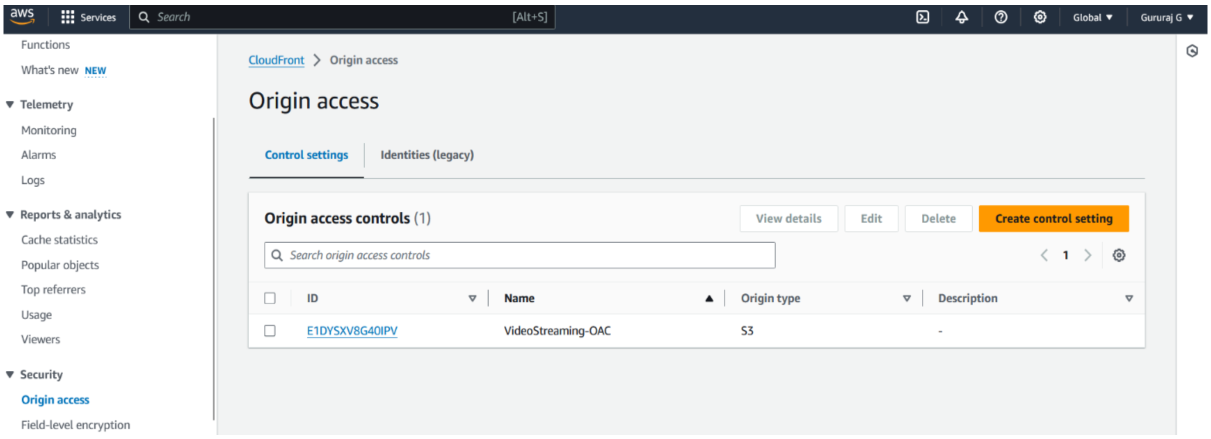


Distributions (1)

ID	Descripti...	Type	Domain name	Alternat...	Origins
E1MKD664IYIR1E	-	Production	d21rr2y0u23ont.cloudfront.net	-	guru-videostorag

Step 6: After Opened it,
Go left to Console panel,*::* and select ***Security.***

Step 7: In ***Security***, Select ***Origin Access*** and then, In ***Create Control Setting.***



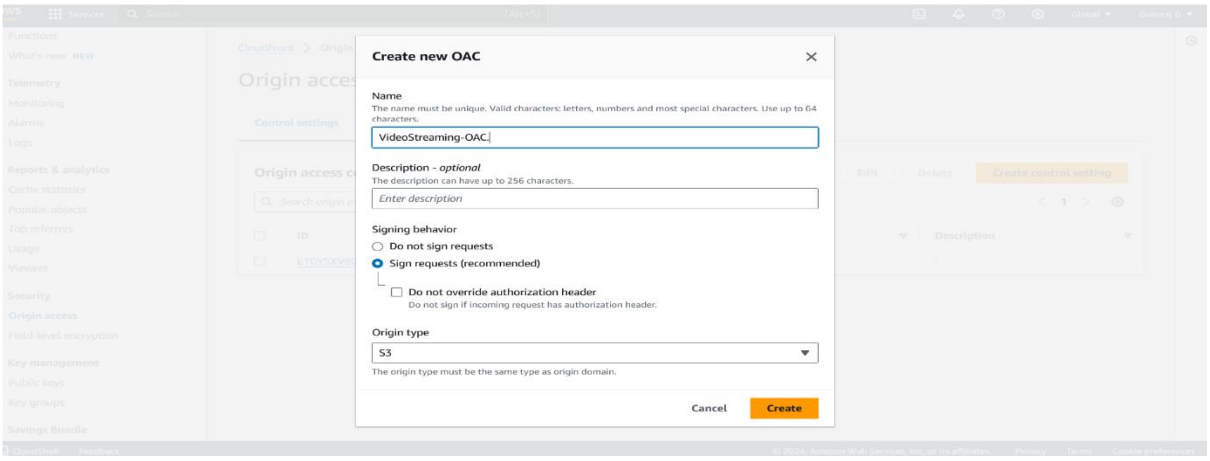
Origin access

Origin access controls (1)

ID	Name	Origin type	Description
E1DYSXV8G40IPV	VideoStreaming-OAC	S3	-

- In the, ***Details*** field, In the, ***name*** field Give
- name, example: ***VideoStreaming-OAC.***

- Keep remaining things as it is and ***check*** in the
- filed ***Settings, Sign Request*** is ***selected or not.***
- If it is ***selected,*** keep as it is, ***otherwise*** select it.



Create new OAC

Name
The name must be unique. Valid characters: letters, numbers and most special characters. Use up to 64 characters.
VideoStreaming-OAC.

Description - *optional*
The description can have up to 256 characters.
Enter description

Signing behavior
○ Do not sign requests
● Sign requests (recommended)
☐ Do not override authorization header
Do not sign if incoming request has authorization header.

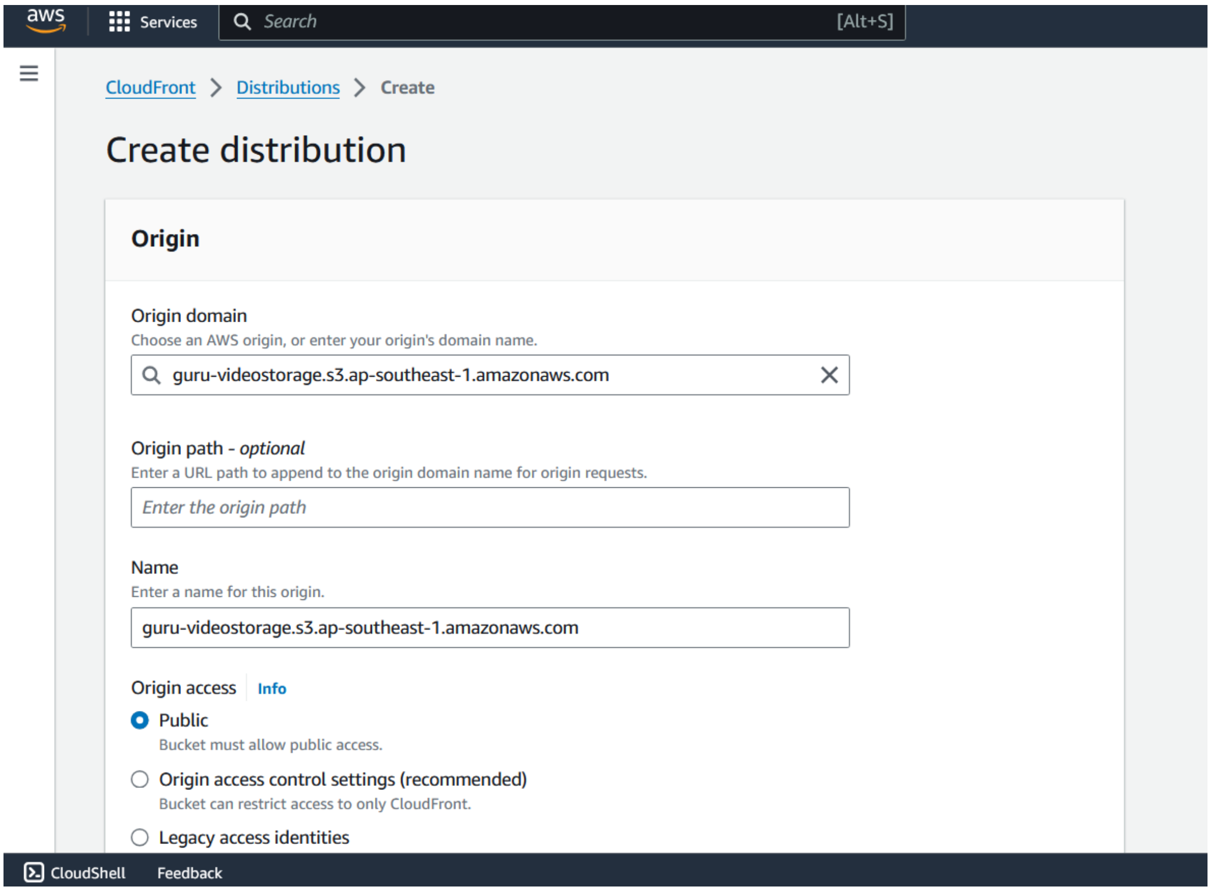
Origin type
S3
The origin type must be the same type as origin domain.

- Keep remaining things as it is and ***check*** in the
- filed ***Settings, Sign Request*** is ***selected or not.***
- If it is ***selected,*** keep as it is, ***otherwise*** select it.

Step 8: Select, ***Create.***
Finally, Successfully, created.

Step 9: Goto, the ***Distribution*** Filed and ***Create a distribution.***

Step 10: In the, ***Create Distribution***, ***Origin Filed***,
Select, ***Origin domain*** as your ***Created Domain*** which relates to ***video storage***.



CloudFront > Distributions > Create

Create distribution

Origin

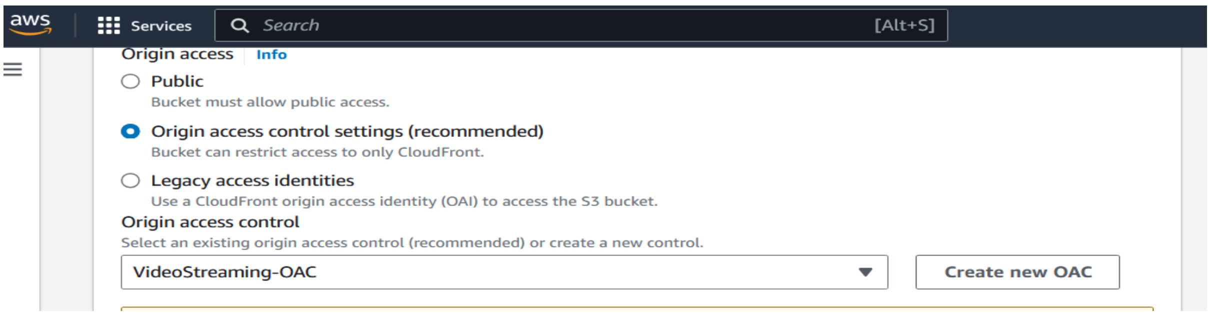
Origin domain
Choose an AWS origin, or enter your origin's domain name.
guru-videostorage.s3.ap-southeast-1.amazonaws.com

Origin path - *optional*
Enter a URL path to append to the origin domain name for origin requests.
Enter the origin path

Name
Enter a name for this origin.
guru-videostorage.s3.ap-southeast-1.amazonaws.com

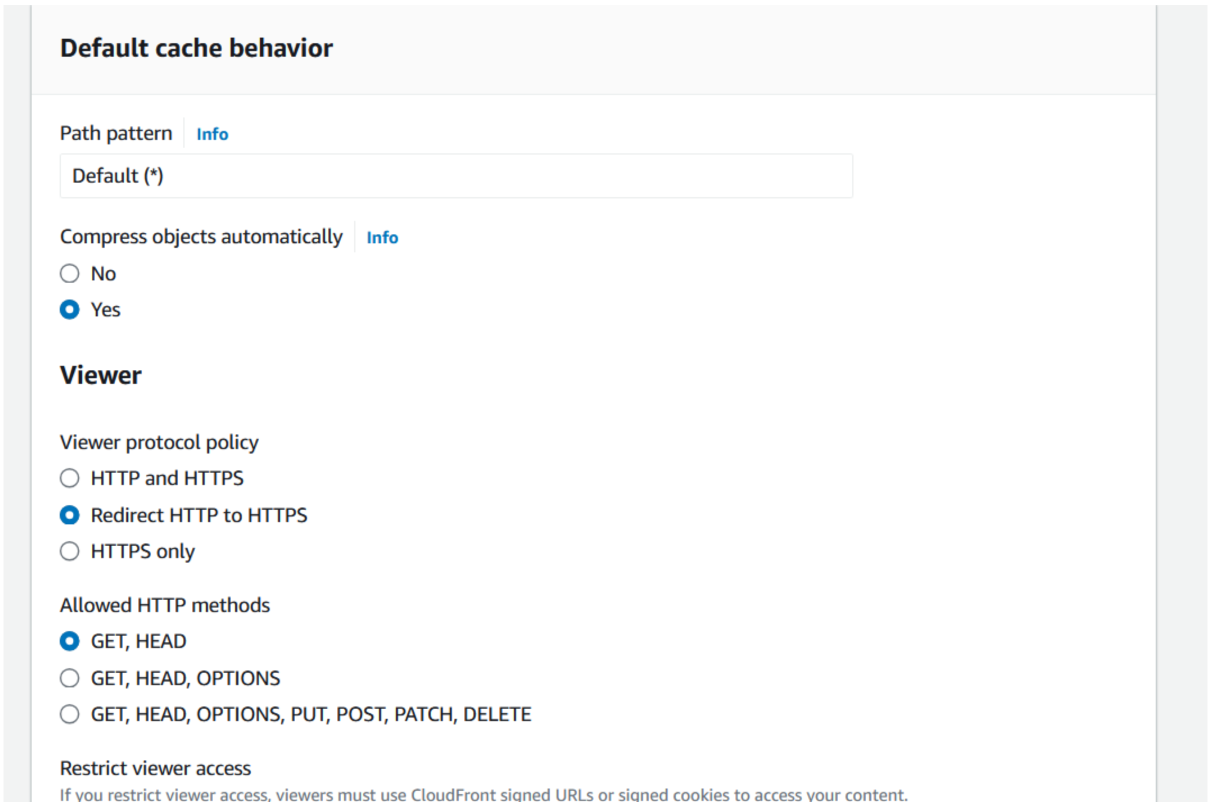
Origin access Info
[✓ Public]
Bucket must allow public access.
Origin access control settings (recommended)
Bucket can restrict access to only CloudFront.
Legacy access identities

- In the, ***Origin access***, Select, **Origin Access Control setting.** Which is, we created, ***VideoStreaming-OAC.***



Origin access Info
Public
Bucket must allow public access.
[✓ Origin access control settings (recommended)]
Bucket can restrict access to only CloudFront.
Legacy access identities
Use a CloudFront origin access identity (OAI) to access the S3 bucket.
Origin access control
Select an existing origin access control (recommended) or create a new control.
VideoStreaming-OAC

- **Scroll Down,** In ***Default cache behavior*** field, In the ***viewer***, ***View Protocol Policy***, Select ***Redirect Http to Https.***



Default cache behavior

Path pattern Info

Default (*)

Compress objects automatically Info

No [✓ Yes]

Viewer

Viewer protocol policy

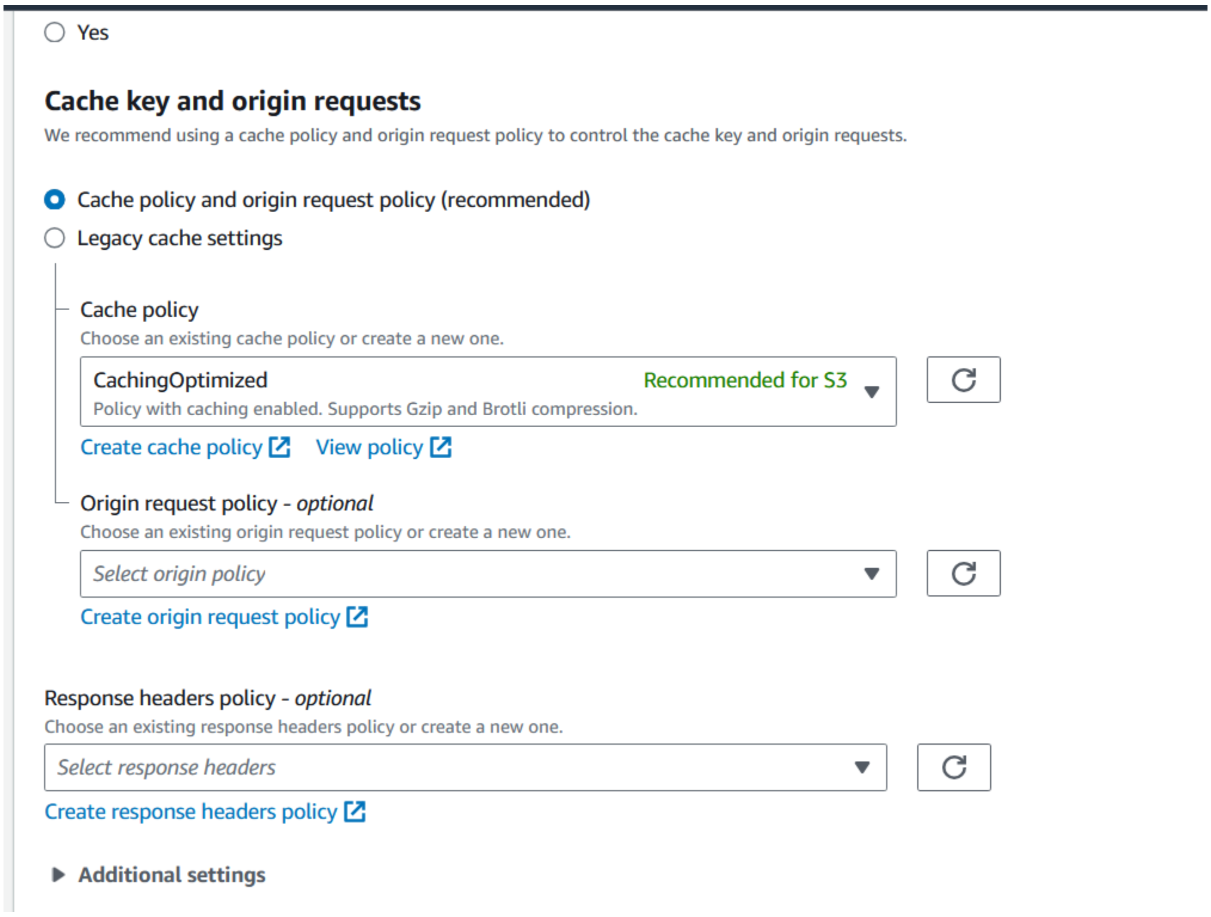
HTTP and HTTPS [✓ Redirect HTTP to HTTPS] HTTPS only

Allowed HTTP methods

[✓ GET, HEAD] GET, HEAD, OPTIONS GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE

Restrict viewer access

If you restrict viewer access, viewers must use CloudFront signed URLs or signed cookies to access your content.



Yes

Cache key and origin requests

We recommend using a cache policy and origin request policy to control the cache key and origin requests.

[✓ Cache policy and origin request policy (recommended)] Legacy cache settings

Cache policy

Choose an existing cache policy or create a new one.

CachingOptimized Recommended for S3

Policy with caching enabled. Supports Gzip and Brotli compression.

Create cache policy View policy

Origin request policy - *optional*

Choose an existing origin request policy or create a new one.

Select origin policy

Create origin request policy

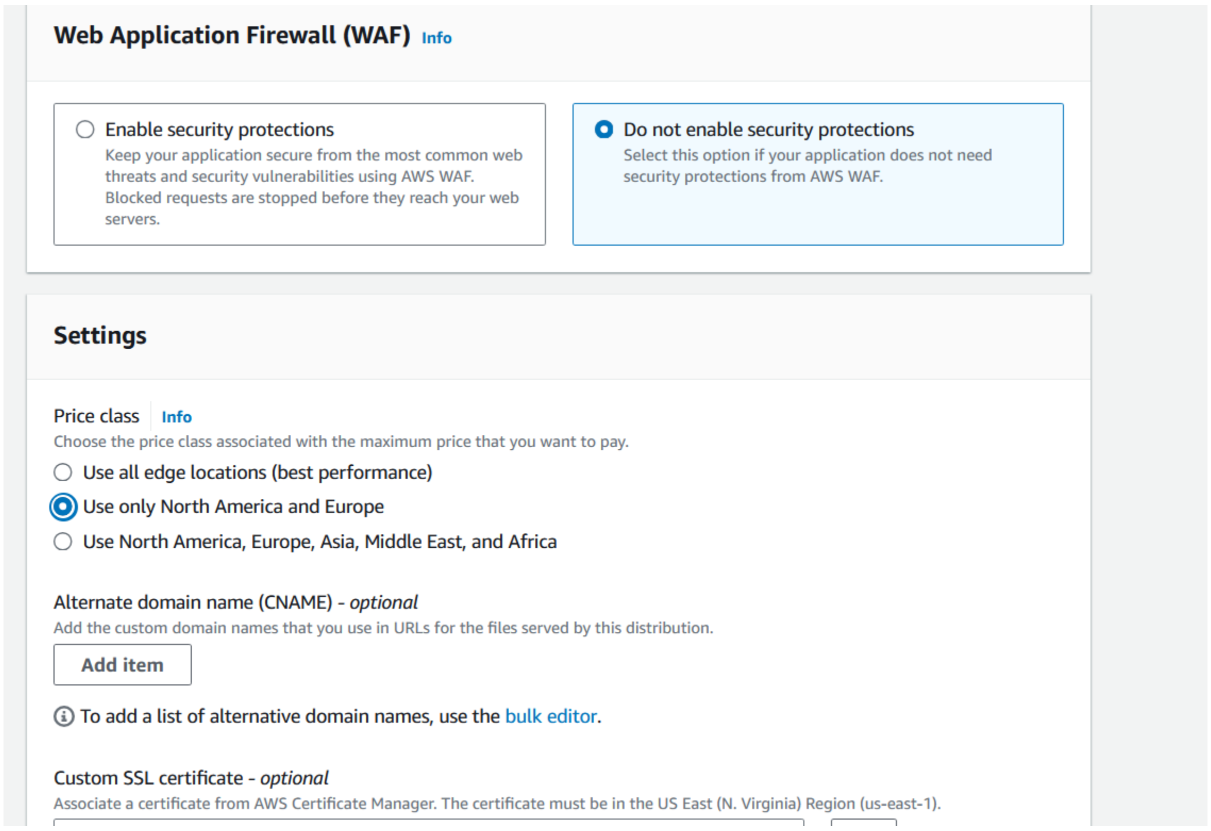
Response headers policy - *optional*

Choose an existing response headers policy or create a new one.

Select response headers

Create response headers policy

Additional settings



Web Application Firewall (WAF) Info

[Enable security protections
Keep your application secure from the most common web threats and security vulnerabilities using AWS WAF. Blocked requests are stopped before they reach your web servers.]

[✓ Do not enable security protections
Select this option if your application does not need security protections from AWS WAF.]

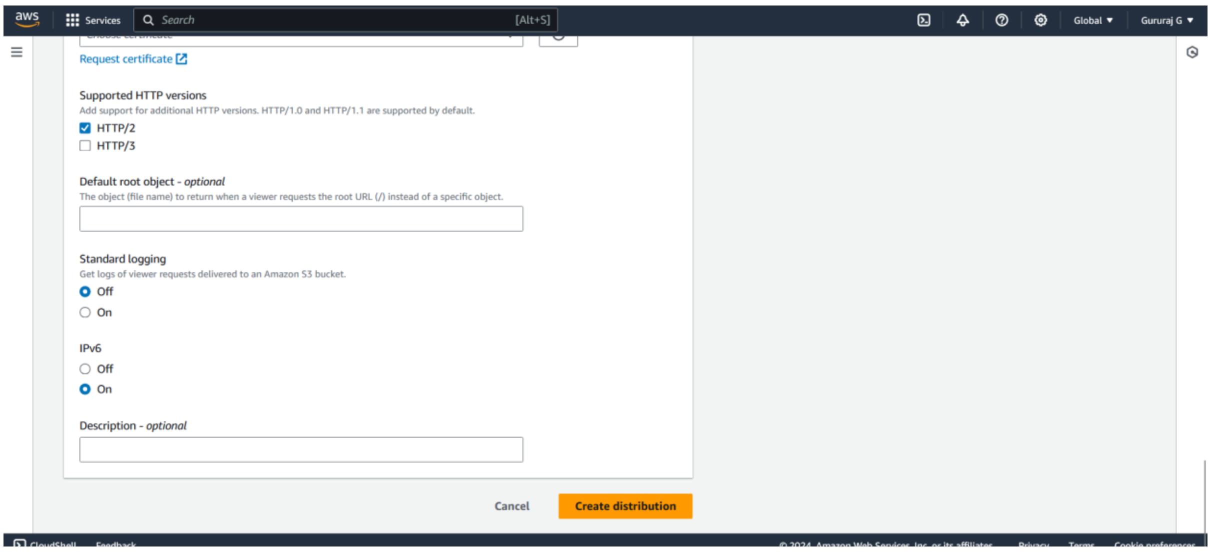
Settings

Price class Info
Choose the price class associated with the maximum price that you want to pay.
- Use all edge locations (best performance)
- [✓ Use only North America and Europe]
- Use North America, Europe, Asia, Middle East, and Africa

Alternate domain name (CNAME) - *optional*
Add the custom domain names that you use in URLs for the files served by this distribution.

To add a list of alternative domain names, use the bulk editor.

Custom SSL certificate - *optional*
Associate a certificate from AWS Certificate Manager. The certificate must be in the US East (N. Virginia) Region (us-east-1).



Supported HTTP versions
Add support for additional HTTP versions. HTTP/1.0 and HTTP/1.1 are supported by default.
- [✓ HTTP/2]
- HTTP/3

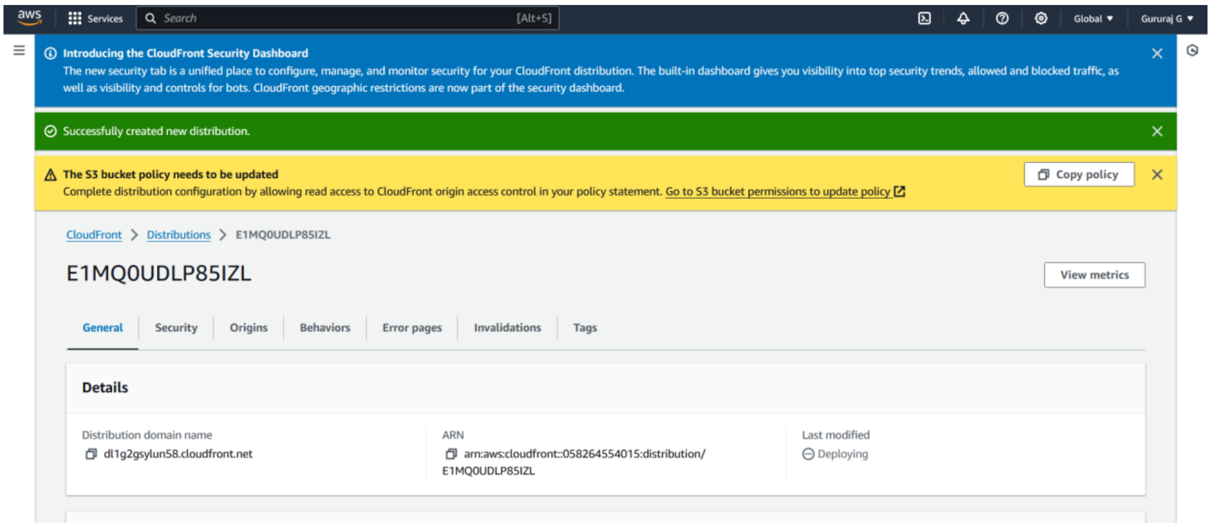
Default root object - *optional*
The object (file name) to return when a viewer requests the root URL (/) instead of a specific object.

Standard logging
Get logs of viewer requests delivered to an Amazon S3 bucket.
- [✓ Off]
- On

IPv6
- Off
- [✓ On]

Description - *optional*

Step 11: Select **Create**,



Introducing the CloudFront Security Dashboard
The new security tab is a unified place to configure, manage, and monitor security for your CloudFront distribution. The built-in dashboard gives you visibility into top security trends, allowed and blocked traffic, as well as visibility and controls for bots. CloudFront geographic restrictions are now part of the security dashboard.

Successfully created new distribution.

The S3 bucket policy needs to be updated
Complete distribution configuration by allowing read access to CloudFront origin access control in your policy statement. Go to S3 bucket permissions to update policy

E1MQOUDLP85IZL

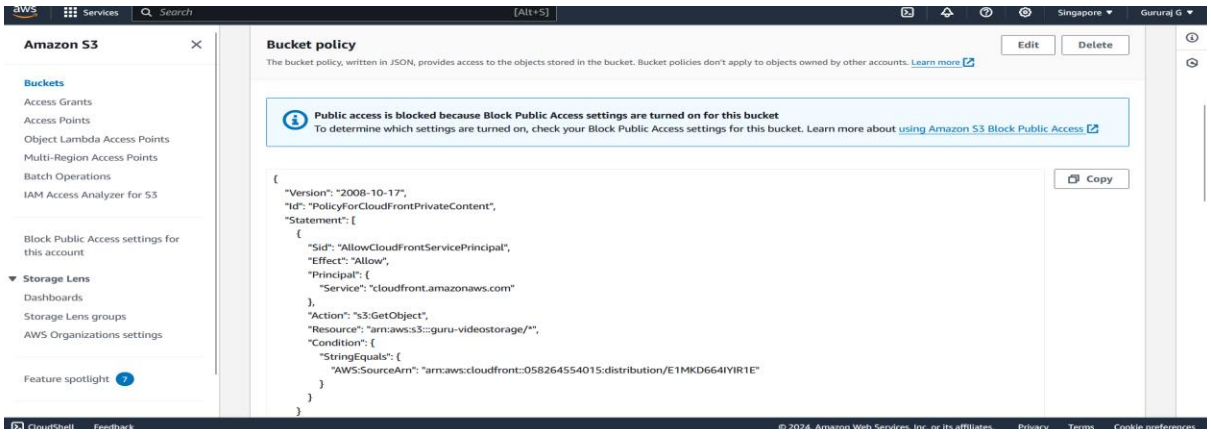
Details

Distribution domain name: dl1g2gsylun58.cloudfront.net
ARN: arn:aws:cloudfront::058264554015:distribution/E1MQOUDLP85IZL
Last modified: Deploying

Finally, Distribution is Created. It takes some times. Above the ***screen*** one popup will come which is related to s3 ***Bucket Policy Copy.***

```
{
        "Version": "2008-10-17",
        "Id": "PolicyForCloudFrontPrivateContent",
        "Statement": [
            {
                "Sid": "AllowCloudFrontServicePrincipal",
                "Effect": "Allow",
                "Principal": {
                    "Service": "cloudfront.amazonaws.com"
                },
                "Action": "s3:GetObject",
                "Resource": "arn:aws:s3:::guru-videostorage/*",
                "Condition": {
                    "StringEquals": {
                      "AWS:SourceArn":
"arn:aws:cloudfront::058264554015:distribution/E1MQ0UDLP85IZL"
                    }
                }
            }
        ]
    }
```

that and ***select Go to S3 Bucket Permission*** and In the bucket **Policy click edit and Paste Policy** and save It.

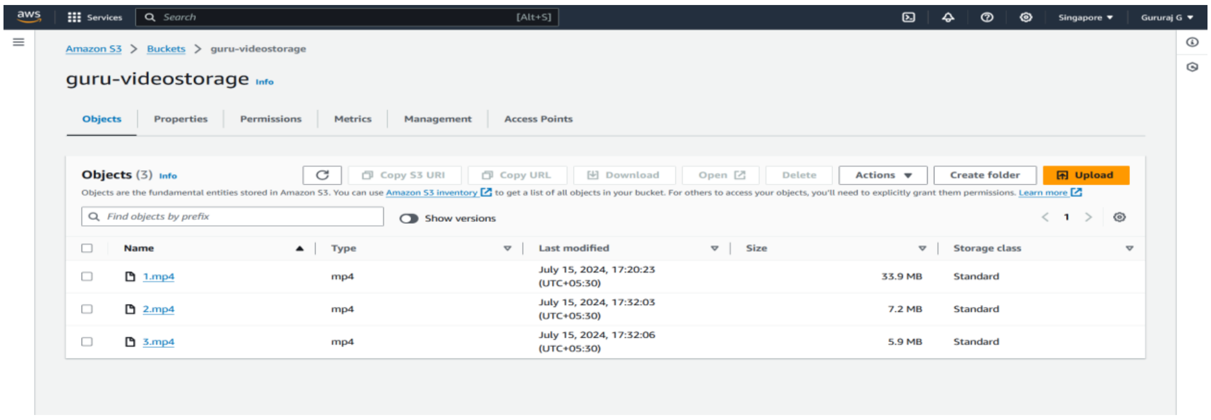

Bucket policy

The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more

Public access is blocked because Block Public Access settings are turned on for this bucket
To determine which settings are turned on, check your Block Public Access settings for this bucket. Learn more about using Amazon S3 Block Public Access

```
{
  "Version": "2008-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::guru-videostorage/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn": "arn:aws:cloudfront::058264554015:distribution/E1MKD664IYIR1E"
        }
      }
    }
```

Step 12: Goto ***S3 and Upload video.***


Amazon S3 > Buckets > guru-videostorage

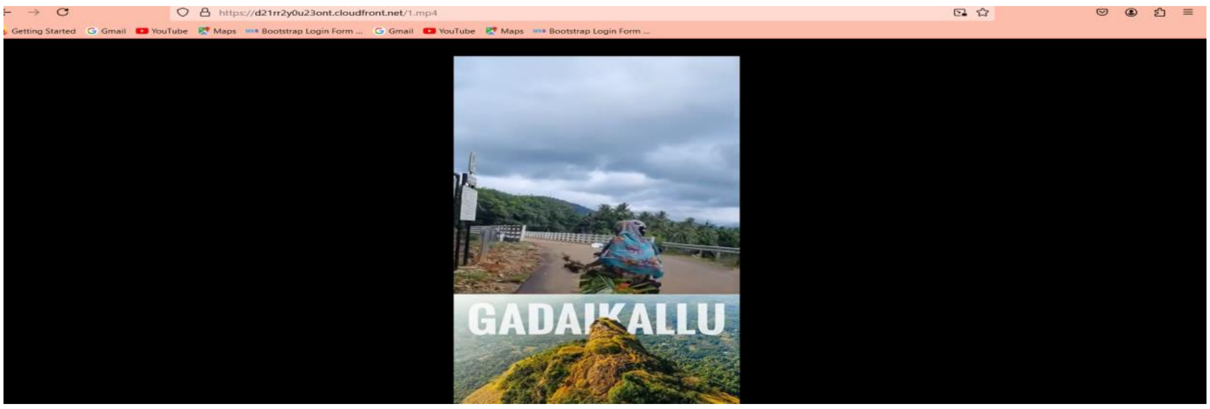
guru-videostorage

Objects (3)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

Name	Type	Last modified	Size	Storage class
1.mp4	mp4	July 15, 2024, 17:20:23 (UTC+05:30)	33.9 MB	Standard
2.mp4	mp4	July 15, 2024, 17:32:03 (UTC+05:30)	7.2 MB	Standard
3.mp4	mp4	July 15, 2024, 17:32:06 (UTC+05:30)	5.9 MB	Standard

Step 13: ***Goto Distribution and copy the url and goto s3 and select video file name with extension and paste distribution url on chrome and add / along with video file name. it will run successfully. And copy url and paste it to your front video src***.



Step 14: video streaming.

Step 15: Copy the url and link this in your react App

```
<source src="https://d21rr2y0u23ont.cloudfront.net/1.mp4" type="video/mp4" />
```

Step 16: Run your app it will sucessfully run.

Type Command:
```
npx create-react-app video-streaming-service
cd video-streaming-service
npm start
```